SLA Specification

Abstract
The current document proposes a Service Level Agreement (SLA) specification in the framework of the SLALOM project. The specification emerges from the analysis of a set of metrics (Service Level Objectives) that have been expressed as of major importance both from cloud providers and from cloud adopters. In this context, the current document focuses on three (3) specific metrics, namely: Availability, Elasticity, and Response Time. The proposed specification is based on an initial approach that enables the specification of "any" metric, through an abstract generic definition. Based on this definition, representative examples are provided for the aforementioned metrics for four (4) different service classes / categories, spanning the infrastructure and service layers of the cloud SPI model: (infrastructure layer) Computational service (i.e. VMs), Networking service, Storage service, and (software layer) application / software service.

Dissemination Level: Public

Release Date: 21 September 2015

The SLALOM Project is co-funded by the European Commission through the H2020 Programme under Grant Agreement 644720

Introduction

In the service provisioning domain, Service Level Agreements (SLAs) serve as the foundation for the expected quality level of the service between the consumer and the provider, the Quality of Service (QoS) given that as "agreements", SLAs encapsulate a set of terms (i.e. metrics) that capture these QoS parameters. However, in the cloud computing domain, the diversity of the proposed SLAs by providers (with marginal overlaps), has led to multiple different definitions / specification of cloud SLAs – each one developed by the corresponding provider.

In this context, SLALOM EU project[1] aims at providing a specification for cloud SLAs that will allow the concise and clear description of SLAs in terms of the defined service quality attributes (objectives) and metrics. The purpose of this document is to serve as a starting point for the development of such a "standardised" SLA. To this end, an abstract formal definition of a "metric" is proposed, so as to provide the ground for a generic, yet uniform, definition of metrics. Based on this definition, representative examples for different metrics (i.e. availability, elasticity and response time) are provided for different cloud services (i.e. computational, networking, storage, software). There should be noted that the basis for this document is a complete analysis of different metrics, their prioritization according to the views of cloud providers and cloud adopters, as well as an analysis of commercial SLAs as well as of research outcomes, which are included in the corresponding SLALOM SLA specification and reference model deliverable[2]. The latter also defines the different elements of an SLA (i.e. parameters, rules, dependencies) which are proposed as a set of elements that define an SLA in a complete way. The representative examples that are cited in this report do not provide the full specification (in terms of fields / elements of an SLA) but focus on the core rule expression for a metric, which is the fundamental element in a cloud SLA.

Definition of abstract metric

The aim of the proposed definition is to enable any cloud provider to define a metric that will be included in the SLA. Given that in order to define and evaluate a metric, a set of samples against its validity are required, the proposed definition is directly linked to these samples. The inclusion of samples is strongly proposed in order to ensure that the metric is both clearly defined and can be evaluated with respect to its fulfilment. The latter has been inspired by the ambiguity that emerges from various existing SLAs such as Amazon EC2[3] and Google Compute[4]. Based on the above, the abstract metric definition is achieved through three (3) individual levels / definitions: sample definition, boundary period and error definition, and abstract metric definition.

Sample definition

The aim of this level / definition is to enable the identification of the samples that satisfy a criterion related to their success. For example if availability is defined in a storage service not only in terms of the success of the operation but also with relation to performance aspects (e.g. GET operation of an object within "x" seconds), a success sample is the one for which the service responds within the "acceptable" time limit. Given that the samples are both of different nature and can be obtained through different mechanisms / means, the SLA specification (defined in the SLALOM SLA specification document in detail) should also include a "field / element" (in the "Rule Definitions" block) that concretizes the

[1] SLALOM EU Project, http://slalom-project.eu
[2] SLALOM Public Deliverable, D3.2 – SLA specification and reference model, http://slalom-project.eu/content/d32-%E2%80%93-sla-specification-and-reference-model
[3] Amazon EC2 Service Level Agreement, http://aws.amazon.com/es/ec2/sla/
[4] Google Compute Engine Service Level Agreement (SLA), https://cloud.google.com/compute/sla

sampling process. This field, namely "*Type of operation*" will refer to the corresponding nature of the process. The following notation is used:

- *Sample Condition - sc*: the condition stating whether a sample has been successful.
  - operator: the operator can either be a boolean one (i.e. AND, OR, NOT) or a comparison operator (<, >, <=, >=, ==, !=).
  - value: the actual value of the condition that can be arithmetic, non-arithmetic (e.g. a string such as "exception") or an enumeration (e.g. HTTP response code == 200).
  - unit: the unit for the value of the condition.
- *Sample - s*: the sample used to evaluate a parameter against the condition sc.
- *Successful Sample - ss*: the sample satisfying the condition sc.
- *Unsuccessful Sample - us*: the sample not satisfying the condition sc.

Sample definition

For a given type of operation as specified in the corresponding field (described previously)

$sc = operator + value + unit$

ss = s if (sc is true)

us = s if (sc is false)

Boundary period and error definitions

The aim of this level / definition is to enable the definition of the boundary period and error for which the analysis of a parameter (through samples) is considered valid. The boundary period is the case for several providers today – for example Google sets a boundary condition to consider a downtime period as actual downtime if it is larger than 5 consecutive minutes[5]. The same applies for error conditions. The overall goal of this level / definition is to identify the set of periods that are "valid" (as successful or unsuccessful) and should be included in the metric definition, based on the individual samples and the required error rate. The following notation is used:

- *Boundary Period - bp*: the period for which the analysis of a parameter (through samples) should be taken into account. Any sample that is not meeting this criterion (i.e. falls within the period) is excluded even though if it is successful (i.e. ss according to the sample definition).
  - operator: a comparison operator (<, >, <=, >=, ==, !=).
  - value: the actual arithmetic value of the condition.
  - unit: the unit in this case is always a time unit (e.g. seconds, minutes, etc).
- *Error Condition - ec*: the error condition ratio for which the analysis of a parameter (through samples) should be taken into account. The ratio is always expressed in a percentage (%) format.
  - operator: a comparison operator (<, >, <=, >=, ==, !=).
  - value: the actual arithmetic value of the condition.
- *Error Ratio - er*: the error ratio calculated based on the total set of samples and the successful samples.
- *Period - p*: the period in which samples (*sc* and *uc*) are examined according to the boundary period and the error condition.
- *Valid Period - vp*: the period for which the error ratio value meets the error condition ratio and the boundary period condition is also satisfied.
- *Non-valid Period - np*: the period for which the error ratio value does not meet the error condition ratio (the boundary period condition is satisfied).

[5] Google App Engine SLA, https://cloud.google.com/appengine/sla

Boundary period and error definitions

$bp = operator + value + unit$

$ec = operator + value + \%$

$er = \Sigma us / \Sigma s \; \forall us \in p$

$vp = p$ if $((er <= ec) \;\&\&\; (p >= bp))$

$np = p$ if $((er >= ec) \;\&\&\; (p >= bp))$

Abstract metric definition

The aim of this level / definition is to provide an abstract format enabling cloud providers to define an SLA metric. While the definition is performed through a condition, a proposal is also provided linking the metric definition with each formal evaluation. The following notation is used:

- *Metric Condition - mc*: the condition regarding a specific metric. The condition is always expressed in a percentage (%) format to enable its evaluation as proposed through the metric evaluation.
    - operator: a comparison operator (<, >, <=, >=, ==, !=).
    - value: the actual arithmetic value of the condition.
- *Metric Evaluation - me*: the evaluation of the metric based on the valid and non-valid period samples. The evaluation should be smaller than the condition (i.e. me <mc).

Abstract metric definition

$mc = operator + value + \%$

$me = \Sigma np \; / \; (\Sigma vp + \Sigma np)$

Representative commercial examples

This section provides representative commercial examples of SLAs based on the aforementioned definitions in order to depict that the generic abstract definition can be adapted to different cases, providers and service types. There should be noted that for all examples the expressions that show validity (i.e. non violation) of the SLA are cited.

Microsoft Azure Storage		
Level / definition	Expression	Notes
Sample definition	sc = 2 sec	Several sampling conditions are defined per type of operation. For example it is specified (exact wording) "*Sixty (60) seconds*" for PutBlockList and GetBlockList.
	Type of operation: PutBlockList and GetBlockList	Several type of operations are defined. An example is provided here.
Boundary period and error definitions	bp > 3600 sec	The exact wording is "*given one-hour interval*".
	ec > 0%	Error condition reflecting that all periods should be taken into account for the availability metric evaluation (exact wording) "*is the sum of Error Rates for each hour*".
Abstract metric definition	availability < 99.9 %	Availability metric definition given the boundary period and error condition.

Amazon EC2		
Level / definition	Expression	Notes
Sample definition	sc: *UNDEFINED*	The sampling condition is not defined in the Amazon EC2 SLA. The concrete wording is "*when all of your running instances have no external connectivity*". Nonetheless, the way to specify / measure "external connectivity" is not defined. For example a customer could use a ping operation or a custom monitoring mechanism.
	Type of operation: *UNDEFINED*	Not defined how the condition of connectivity can be actually measured (e.g. the ping operation mentioned previously).
Boundary period and error definitions	bp > 60 sec	The exact wording is "the percentage of minutes", thus the period is 60 seconds.
	ec = 100%	Error condition reflecting that the error ratio is that for the entire bp the resource must be continuously "*unavailable*".
Abstract metric definition	availability < 99.95 %	Availability metric definition given the boundary period and error condition.

Google AppEngine Datastore		
Level / definition	Expression	Notes
Sample definition	sc: INTERNAL_ERROR	Several sampling conditions are defined per type of operation. For example it is specified (exact wording) "*INTERNAL_ERROR, TIMEOUT, ...*" for API calls.
	Type of operation: API calls	Several type of operations are defined. An example is provided here.
Boundary period and error definitions	bp > 300 sec	The exact wording is "*five consecutive minutes*".
	ec > 10%	Error condition reflecting that the error ratio is (exact wording) "*ten percent Error Rate*".
Abstract metric definition	availability < 99.95 %	Availability metric definition given the boundary period and error condition.

Additional examples

The purpose of this section is to depict the wide applicability of the proposed approach for the definition of SLA metrics both for different service classes (such as computational, storage and software services) and for different metrics (such as availability, elasticity and response time).

Availability for storage service		
Level / definition	Expression	Notes
Sample definition	sc <= 100 msec	Samples regarding availability obtained for example through ping operations to the corresponding hosts. Successful samples are the ones for which ping responds with less than 100 msec (above 100msec or "unreachable" are considered unsuccessful samples).
Boundary period and error	bp > 300 sec	Boundary period of 300 secs reflecting that "sporadic" unavailability (based on the *sc*) will not be counted as actual unavailability periods.

definitions	latency error ratio < 1%	Error condition ratio reflecting the number of cases for which latency (i.e. time for a single I/O operation) cannot exceed the specified value in the dependencyExpression SLA field (e.g. 50 msec).
Abstract metric definition	availability < 99.98 %	Metric definition with respect to availability given the boundary period and error condition (to be considered for the validation of the given availability constraint).

Elasticity for computational service		
Level / definition	Expression	Notes
Sample definition	sc <= 1 min	Samples regarding how fast the provider responds to requests for re-allocation of resources.
Boundary period and error definitions	bp > 10 min	Boundary period reflecting that non-allocation of some resources within 10 mins will not be counted as non-elasticity.
	ec < 5%	Error condition (precision) reflecting the number of resources deployed versus the actually needed ones.
Abstract metric definition	elasticity < 90 %	Metric definition with respect to elasticity given the boundary period and error condition.

Response time for software service		
Level / definition	Expression	Notes
Sample definition	sc <= 1 sec	Samples regarding response time obtained for through different requests (e.g. sequential, parallel, from different locations, etc). Either one or more than one sample conditions can be defined.
Boundary period and error definitions	bp < 30 sec	Boundary period of 30 secs reflecting for example the HTTP timeout period, within which requests not accommodated will not be counted as actual non-responsiveness.
	ec < 7%	Error condition (response) reflecting the number of cases for which the response time cannot exceed the specified value of the *sc*.
Abstract metric definition	response time < 97.77 %	Metric definition with respect to availability given the boundary period and error condition (to be considered for the validation of the given availability constraint).

Conclusions

This document summarizes the SLALOM SLA specification with respect to a formal definition of an SLA metric along with the definition of additional attributes that can be used for its validation. A complete specification is provided in the corresponding SLALOM deliverable, thus this report only focuses on the definition, which should be placed in the identified "placeholders" of the SLA (i.e. rule and dependency expression fields of the SLA). Note that a new field, namely "Type of operation" in the "Rule Definitions" block is proposed, which concretizes the sampling process. Based on the feedback obtained by the stakeholders – cloud providers and adopters, an updated version both of the complete specification and of this report will be released within 2015.